**Technical assignment:** CRUD application using Spring boot and Maven.

**Description:**

Your challenge is to create a CRUD application using Spring boot and Maven that can store messages submitted via API in the form of text files.

The app must have (at least) one "Message" entity that can be interacted with via REST API.

The Message entity should contain (at least) the fields “id” and "text", which stores the content of message.

  - GET requests should access any existing messages or a specific message by ID and should return data in JSON format

  - POST requests should accept any text file as a parameter and create a new Message entity entry from the content contained in the file

    e.g.

   \*\*test-file.txt\*\*

    ``` txt

    hello, world!

    ```

  - PUT requests should accept JSON object to update existing entries

  - Data validation: The application should enforce a 255-character limit for the message text.

For persistence, you can use an in-memory database such as H2.

Try to incorporate as much of the following as possible in your architecture:

  - Annotation-based configuration

  - Spring data repositories

  - Data transfer objects

  - Service layer

**Solution:**

1. First I have generated project using <https://start.spring.io/> link with dependencies Spring WEB, Spring Data JPA, H2 Database, Spring Boot DevTools added
2. Saved the generated zip file and unzip it
3. In Eclipse IDE import ->maven ->Existing Maven projects ->point to root directory=<path of the directory where zip file is extracted>
4. All the java files I have created to implement the “Spring Boot CRUD application” are uploaded in my GitHub repository. Here is the link https://github.com/pskadam23/SpringBootCRUDApp

**Testing:**

Run the application in Eclipse IDE and test the application from command line prompt by running following commands:

1) Display all messages:

curl -v localhost:8080/messages

2) Read message text from a text file and saves as new message. If the message text length exceeds the limit of 255 characters then error message is displayed

curl -X POST localhost:8080/messages -H "Content-type:application/json" -d "{\"text\": \"C:\\tmpFolder\\test.txt\"}"

3) Update existing message text if the message is present with the given Id and the message length doesn't exceed the limit of 255 characters.

If message with given Id is not present or if the message text length exceeds the limit of 255 characters then error message is displayed.

curl -X PUT localhost:8080/messages/3 -H "Content-type:application/json" -d "{\"text\": \"updated message\"}"

4) Delete message with given Id.

curl -X DELETE localhost:8080/messages/3

**Notes:**

By default, Spring Boot configures the application to connect to an in-memory store with the username sa and an empty password.

Spring Boot’s embedded Apache Tomcat server is acting as a webserver and is listening for requests on localhost port 8080.